eChallenges e-2009 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2009
ISBN: 978-1-905824-13-7

Semantic Mediation to Enable Network and Service Management Interoperability in Future Internet Networks

John STRASSNER[1,2], Sven VAN DER MEER[1], Brendan JENNINGS[1],
Miguel PONCE DE LEON[1]
[1]*Telecommunications Software & Systems Group, Waterford, Ireland*
Tel: +353 51 302910, Fax: + 353 51 341100,
Email: {jstrassner, vdmeer, bjennings, miguelpdl}@tssg.org
[2]*Pohang University of Science and Technology, Pohang, Korea*
Tel: +82 54 279 5605, Fax: + 82 54 279 5699, Email: johns@postech.ac.kr

Abstract: Despite its success, the current Internet considered management only as an afterthought, and was mainly limited to device monitoring and configuration management. Management has been inhibited by the proliferation of vendor-specific data models and programming languages. Future Internet management will be even more demanding due to the large increase in the number of users, services, and other requirements. This paper introduces a novel knowledge representation mechanism that enables the mapping of syntax and *semantics* to different vendor-specific devices so that their diverse behaviour can be orchestrated. Our architecture has been deployed, and enables clean-slate mechanisms to be deployed within a framework that supports backwards compatibility with current and legacy devices.

1. Introduction and Objectives

The current success of the Internet architecture has spurred advances in business, social, and technical communications. However, the simplicity of the original Internet architecture is also the source of many of its inherent limitations [1]-[3]. Two of the most important are its architectural limitations and its inability to relate business needs to network services and resources offered. While the former problem is being worked on across the world [4]-[9], the latter problem tends to be ignored. One of the main reasons for this is that current network management and operational data from devices does not contain business or system information. This means that business problems must be *inferred*. Each vendor uses different programming languages and models with different semantics, making this a very difficult problem that impedes the interoperability of end-to-end solutions.

This paper describes a novel representation of knowledge to enable diverse syntax and semantics to be *mapped* to a common internal format for machine-based learning and reasoning. This is used to enhance a proven and deployed autonomic architecture that orchestrates the behaviour of network devices, enabling revolutionary mechanisms to be safely embedded into an evolutionary infrastructure that is based on modern needs. Thus, we provide a deployable framework for clean-slate innovation whilst maintaining backwards compatibility with current and legacy devices, protocols and systems.

2. Methodology

The FOCALE [10] autonomic architecture has been deployed by many industrial and academic institutions. The Inference Plane [11] enhances this architecture by enabling clean-slate approaches to peacefully co-exist within an evolutionary framework that is compatible with existing approaches. The latest version of the DEN-ng information model [12] is currently being standardised in the Autonomic Communications Forum (ACF); previous versions of it have already been standardised in the TeleManagement Forum and in the ITU-T. The DENON-ng ontology [13] and the knowledge representation framework are both mature, and will be standardised by the Autonomic Communications Forum.

We have independently carried out experiments to prove the validity of our concepts; we seek to use this assembly to attract new entities to the Autonomic Communications Forum (ACF), where much of this initial work is being done, and to future FP7 ICT proposals. Our methodology includes interoperability testing in the ACF along with other groups, use of FIRE-compliant testbeds, and prototyping in ongoing FP7 projects.

3. Technology Description

There are three fundamentally different approaches for the design of the Future Internet. The first is an incremental or evolutionary approach, and is evidenced by the plethora of solutions being applied to the current Internet that violate its architectural principles [1][2], such as Network Address Translators, firewalls, and Virtual Private Networks. The second approach, referred to as a “clean slate” approach [3], eliminates existing commitments and starts with a new set of ideas. The third approach is a compromise between these, and enables new ideas to evolve while simultaneously emphasising backwards compatibility with the existing Internet. This is very important to certain stakeholders, such as Internet Service Providers (ISPs), who have invested billions into their equipment and personnel, and want to leverage those investments. It also recognises that current and future networks and networked applications have vastly different requirements; this implies that a single architecture cannot simultaneously meet these different needs.

Most of the current effort in Future Internet development is focused on novel low-level mechanisms, such as different types of routing or protocol stacks. To date, discussions at Future Internet fora and conferences still assume the use of command line interfaces (CLIs) and/or the Simple Network Management Protocol (SNMP) to perform configuration, monitoring, and similar tasks, despite the recognised failure of these two mechanisms [14]. This paper represents the first of a series of innovations to address this problem, and focuses on providing enhanced semantic reasoning.

3.1 Architectural Approach

The purpose of autonomic computing is to manage complexity. The name was chosen to reflect the function of the autonomic nervous system in the human body. By transferring more manual functions to involuntary control, additional resources (human and otherwise) are made available to manage higher-level processes.

The fundamental management element of an autonomic computing architecture is a control loop [1][15]. Sensors retrieve data, which is then analysed to determine if any correction to the managed resource(s) being monitored is needed (e.g., to correct non-optimal, failed or error states). If so, then those corrections are planned, and appropriate actions are executed using effectors that translate commands back to a form that the managed resource(s) can understand. If the autonomic network can perform manual, time-consuming tasks on behalf of the network administrator, such as simple configuration tasks, then that will free the system and the administrator to work together to perform higher-level cognitive functions, such as planning and network optimisation.

The automation of complex, manually-intensive configuration tasks that are prone to error is a primary motivation for autonomic systems. Figure 1 shows a simplified version of our FOCALE *model-driven* autonomic architecture. FOCALE stands for **F**oundation – **O**bserve – **C**ompare – **A**ct – **L**earn – r**E**ason, which describes its novel control loop.

Model-driven means that it can dynamically generate code to (re)configure managed elements from its model [16].

FOCALE uses DEN-ng and DENON-ng to translate disparate sensed data into a common representation. DEN-ng is used to represent facts; DENON-ng is then used to augment these facts with consensual meaning and definitions so that vendor-specific concepts can be mapped into a common terminology, enabling ontology-based inferencing to be used to reason about the system. More importantly, business goals and objectives can be directly related to the system using context-aware policy models [17] that use the Policy Continuum [18] to relate the needs of different constituencies (e.g., business, network, and programming people) to each other. The Policy Continuum enables policies written using terminology and concepts for one domain, such as business analysts, to be translated to policies written using a different set of terminology and concepts for another domain, such as programmers. This enables context-aware policies to be used to orchestrate behaviour for business goals, social networks, and other forms of interaction.

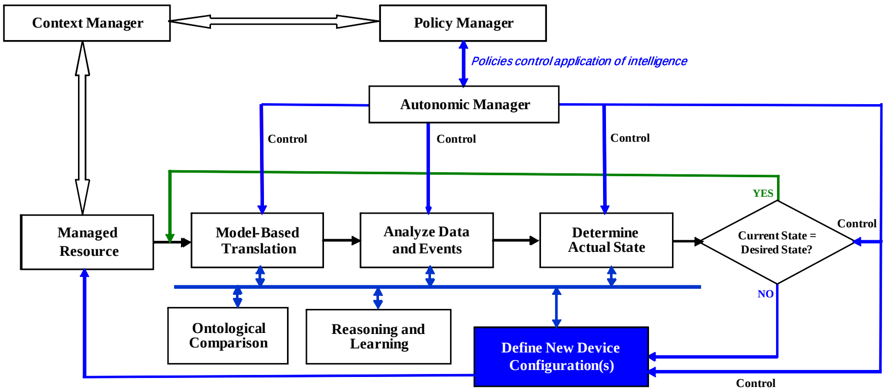

Figure 1. Simplified Version of the FOCALE Autonomic Architecture

The key to FOCALE's adaptive control loops is the interaction between the context manager, policy manager, and autonomic manager. The context manager detects changes in the network, or in the needs of its users; these context changes in turn activate an associated set of policies that define the functionality that the autonomic manager should govern. If no problems are detected, the system continues using the monitoring loop; otherwise, it uses the reconfiguration loop so that the services and resources provided can adapt to these new needs. The autonomic manager uses these policies to govern each of the architectural components of the control loop, enabling the different control loop components to change how it operates as a function of context. Hence, when the sensors detect, or when the autonomic manager *infers*, a context change, then a new set of policies are selected that are applicable to the new context, which changes the functionality provided.

FOCALE develops and uses *libraries* of model and ontology fragments and coded behaviours, much as a library of string processing functions is used by a programming language. Each reusable library object is supported by both models and ontologies. Library behaviours are governed using policy rules, which in turn are selected by a particular context as previously described.

3.2 Knowledge Representation

FOCALE supports a dynamically updateable knowledge base – one that can incorporate new knowledge at runtime as new knowledge is discovered. Our approach supports this requirement by using semantic reasoning to examine input data to see if it is new as well as to determine if it is different (and especially, if it leads to different conclusions) than that already stored in the knowledge base. First order logic (FOL) is used to reason about the validity of the new or changed information with respect to the rest of the knowledge base using the axioms and theories present in the existing knowledge base. Once the new or changed data are determined to be valid, additional logic checks the relationships of the changed data to see if those data also need to be changed. Similarly, existing axioms and theories are applied to the new data to hypothesise new relationships.

Both this and the model-based translation function use the notion of semantic relatedness [19] to determine the relevance as well as the validity of the sensor information as well as inferences derived from those data. Semantic relatedness enables entities that are semantically related using synonymy (e.g., "bank" and "lending institution"), antonymy (e.g., "accept" and "reject"), and other lexical relationships such as meronymy (e.g., court is a part of government), as well as defined associations (e.g., router uses protocol). We use WordNet [20], which provides a set of APIs for computing common linguistic relations and structural matching algorithms, as the basis of our semantic relatedness computation.

3.3 Semantic Mediation

Network management applications must deal with heterogeneous data defined by different languages, as well as be able to combine data from different entities that are not intrinsically related to each other. Our approach builds an extensible reference framework by combining knowledge from information models and ontologies into a rich graph structure, where different knowledge elements are related to each other by dependency and semantic relationships. This enables different types of matching to be performed between received data and the framework, so that received data can be identified and translated into an interoperable form to assemble a complete understanding of the environment. Second, it enables applications to reason about data that is not an exact match so that it can be properly categorised and integrated into the knowledge base. Third, it supports mediation between and negotiation of functionality between disparate entities. For example, if traffic flows through two different vendor devices that use different languages and have different functionality, then suitable configurations must be designed that meet the needs of the traffic in different vendor-specific ways.

We first transform vendor-specific data into a vendor-neutral form. Then, we construct a multi-graph, whose nodes are populated by elements from the DEN-ng information model and the DENON-ng ontology, and whose edges are different types of dependency relationships. We then add semantic relationships to the multi-graph in order tos perform a search to find the highest semantically related match between nodes in the multi-graph.

The semantic content of objects and inter-object relationships is represented in First Order Logic (FOL) and effectively estimated by the complexity of the corresponding FOL expressions. Different aspects of an object, such as its structural components, class references, constraints, and functions, as well as relations between concepts, will be described in FOL as separate logical statements.

We abstract the model of an entity into a number of *aspects*. This enables the complex internal structure of an entity, as well as its participation in different interactions, to be represented in a reusable manner. For example, two different types of routing, such as RIP and OSPF, can be regarded as semantically similar regarding their structural aspects (e.g., both use common concepts, such as source and destination IP address) but dissimilar regarding their behaviour (e.g., they use different algorithms). This enables both a richer evaluation of their semantic relatedness, but also lets an application determine the level of important that each set of aspects contributes. The output of our semantic relatedness algorithm classifies two concepts along a continuum of exactly equal (1.0), exactly not equal (0.0), or partially related ($0.0<i<1.0$). In this latter case, we distinguish between concepts that are (1) partially related but contain some unrelated terms, and (2) subsumed (i.e., concept A is completely contained in concept B).

We then annotate the normalised format with the semantic relatedness values and use this to determine the current state of the entity being managed. The state of a managed entity can be determined by reading appropriate values from its configuration and/or by analysing monitored management data. Each set of configuration changes corresponds to a distinct state. Hence, the managed entity can be related to a state machine, where nodes and edges of the state machine correspond to distinct configurations and changes to those configurations, respectively. This means that orchestration is calculated by finding the best set of state transitions to move the current state to the desired state. By appropriate weighting of the edges of the state machine, different graph algorithms can be used to compute the optimal path.

This is a generic mechanism that can be used to mediate between different commands from different vendors as well as different functionality offered by different devices. Both of these types of mediation can be represented in a graph-theoretic manner; in the first case, each vendor's commands are represented as a graph, and in the second case, each function is represented as a set of classes that can be translated into a graph. In the approach pioneered in FOCALE, each state can have an associated configuration. Orchestration of behaviour is then accomplished by orchestrating the state transitions taken, which controls the set of commands that make up the configuration of a device or set of devices. We use policy rules to determine the cost of an edge, which enables standard graph algorithms to be used to determine the best path (i.e., set of configuration changes) to make. This is shown in Figure 2, where two different policies, P_1 and P_2, are used to determine the cost of the edges e_{ab} and e_{bc}. More specifically, Action A_1 of P_1 and Action A_2 of P_2 are used to set the cost of edges e_{ab} and e_{bc}. Meta-policies can also be used to orchestrate the application of policies to selected edges in the graph.

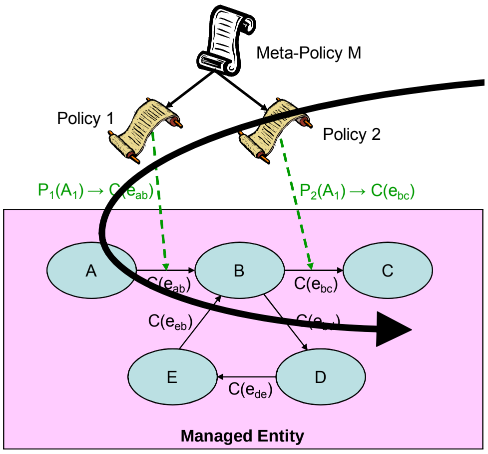

Figure 2. Policy-Based Control of the Cost of an Edge

It is also important to consider semantics that are not able to be represented in network management data. Our approach enables "macro" configuration changes to be represented as a set of "micro" configuration changes using hierarchical state machines. This allows context-sensitive semantics to be used to inspect the outcome of a policy and reason as to whether other changes should be made. For example, in Figure 2, Policy P_1 is used to transition the state of the ManagedEntity from state A to state B. After action A_1 of P_1 is executed, the system can then check (for example) one or more attributes to verify that the ManagedEntity is in state B. If this is true, then the system invokes policy P_2 in order to continue the original reconfiguration; otherwise, a different policy or set of policies will be invoked to fix the problem. In our implementation, we use hybrid reinforcement learning to record which configuration changes are successful and which have problems; we can then analyse the problems to improve their remediation.

3.4 *Context-Aware Semantic Orchestration*

The DEN-ng context-aware policy model [16], shown in Figure 3, is used to model services whose functionality changes in accordance with changes in context. The Context class represents the complete, aggregated context for the entity being managed, and the ContextData class represents a manageable aspect of the complete Context. For example, a communication between two people could model the communication as Context, and use ContextData to model the hardware, software, protocol, environment, and other aspects of the Context for each user. Both Context as well as ContextData can affect the set of policy rules that are currently used to govern behaviour; this enables granular decisions that are dependent on one or more aspects as well as a higher level changes in the overall context to both change policy rules. These two changes can also be linked to the changing of state in one or more FOCALE state machines; this is provided in a two step process. The first step defines the set of policy rules that are used to determine how context affects a state, and the second relates context to state. These are defined by the two associations ContextHasState and ContextDataHasState. The ContextFact and ContextInference classes, along with similar classes for ContextData (not shown in Figure 3), are used as containers to hold the results of applications that perform the operations. This enables FOCALE to use a stable model as a template to generate code, and provide "hooks" for external applications to place their results in the template.

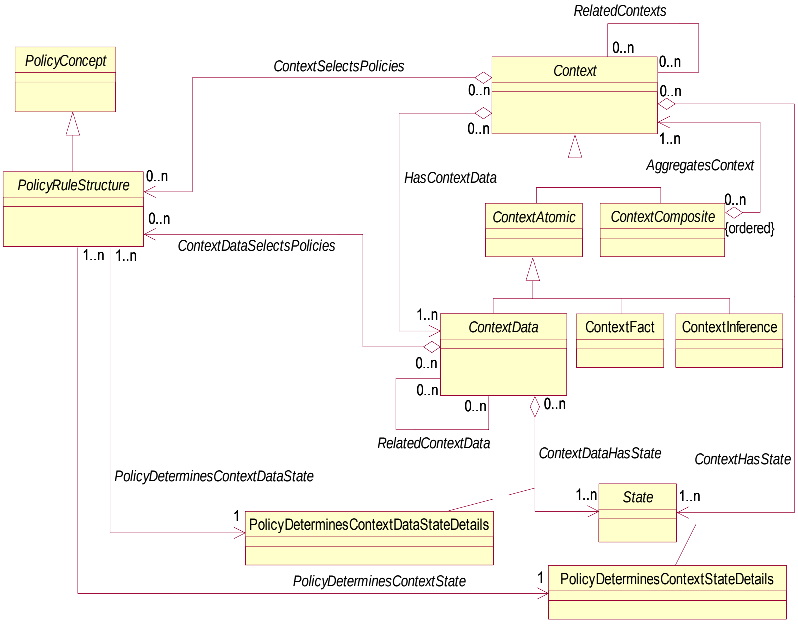

Figure 3. Context-Aware Policy Model

3.5 *Comparison to Other Future Internet Approaches*

Many other approaches that are working on Future Internet activities do not explicitly consider management; of those that do, the main difference between them and our approach is the way that knowledge is represented. We use knowledge to build a set of service-aware networks that can *exchange semantic information to enable their collaboration*. Thus, instead of trying to share fault and other types of data between domains, our approach instead exchanges semantic data describing the *effect* of those data. For example, consider two network devices from different network manufacturers, where each cannot understand the configuration data of the other. Our approach understands what the configuration is being used for, and instead passes the appropriate semantic data such that the FOCALE autonomic manager can generate appropriate vendor-specific configuration data for each.

4. Industrial Significance

We use Rational Rose v7.0.0.0 for information modelling, and Protégé 3.4 beta for ontology design. By transforming existing knowledge from model elements and ontologies into a common format (graphs), it becomes possible to use established linguistic relationships to associate knowledge from model elements with knowledge from ontologies (and vice-versa). This approach lends itself to reusing existing languages, as well as developing new languages. Our algorithms execute in times ranging from less than a minute to less than 30 minutes for small instantiations (tens to hundreds of network objects), showing that this approach can be used for online analysis of local problems. For larger networks, execution time increases with the number of network objects. However, the approach is amenable to parallel processing. Additional testing is required in this area.

This approach is based on FOCALE, which has been deployed by several service providers and enterprises. The enhancements described in this paper provide more robust and additional types of machine-based learning and reasoning to the original FOCALE architecture, and have been validated in closed world scenarios. This enables this approach to be commercialised in a straightforward manner. The main difference between this approach and standard network management systems is that it provides machine-based learning and reasoning that can each be customised to suit the needs of the applications using the system and, more importantly, provide a translation between high-level business needs and low-level network configuration. Therefore, this approach is widely applicable to monitoring, fault and configuration management, and other management operations.

5. Benefits

The main business benefit of our approach is the ability to intelligently automate the behavioural orchestration of network services and resources. This can significantly reduce operational expenditures in network maintenance. Intelligent automation is achieved through (1) generating reusable code from the model, and (2) reasoning about different conditions and requirements in order to choose the best actions to take given unforeseen circumstances. Capital expenditures can also be reduced, as our approach can be used to identify new ways to reuse existing equipment.

From an engineering point of view, our approach enables the modelling and reuse of behaviour. Just as a programmer uses string functions, developers of our approach can define reusable classes and behaviour as libraries; code can then be generated for them or reused from somewhere else, like a vendor’s class library, to implement behaviour. This enables vendor-specific data to be translated into machine-understandable semantics, which enables different parts of an organisation, such as business analysts and network administrators, to better collaborate. It also dramatically simplifies managing heterogeneous devices by providing a common representation of the configuration of the device; this enables machine-based learning and reasoning to be used to automate the configuration management. In addition, our approach can relate these data to business rules and policies, so that each domain will recognise the services and resources that are important to its collaborating domains and treat them accordingly. Finally, our approach defines a single framework that can be re-purposed to support multiple uses and applications.

6. Conclusions

This paper described enhancements to the FOCALE autonomic architecture that provides a new knowledge representation that augments a model-driven architecture with semantics.

This new representation enables semantic mediation to be performed using first order logic, which supports automated behavioural orchestration. This approach has been validated in small test networks to build remediation recommendations based on inferring root cause from sensor data. Future work includes testing against larger networks of more diversified devices, which will involve integrating additional semantic relationships and scaling our existing approach. We will also investigate defining new machine learning algorithms, and standardising elements of our architecture in appropriate fora.

Acknowledgements

This work is partially sponsored by the WCU (World Class University) program through the Korea Science and Engineering Foundation funded by the Ministry of Education, Science and Technology (Project No. R31-2008-000-10100-0). This work is also partially sponsored by Science Foundation Ireland (SFI) under grant number 08/SRC/I1403 (Federated Autonomic Management of End to end services, FAME).

References

[1] D. Clark, et al., “NewArch: Future Generation Internet Architecture”, NewArch Final Technical Report, http://www.isi.edu/newarch/

[2] Blumenthal, M., Clark, D., “Rethinking the design of the Internet: the end to end arguments vs. the brave new world”, ACM Transactions on Internet Technology, Vol. 1, No. 1, pp. 70-109, Aug. 2001

[3] Feldmann, A., “Internet clean-slate design: what and why?,” ACM SIGCOM Computer Communication Review, Vol. 37, Issue 3, July 2007

[4] Future Internet Network Design (FIND), http://find.isi.edu

[5] Global Environment for Network Innovations, http://www.geni.net

[6] http://www.future-internet.eu/

[7] Future Internet Research and Experimentation (FIRE), http://cordis.europa.eu/ist/fet/comms-fire.htm

[8] AKARI, “Architecture Design Project for New Generation Network,” http://akari-project.nict.go.jp/eng/index2.htm

[9] Future Internet Forum (FIF), http://mmlab.snu.ac.kr/fif.

[10] J. Strassner, N. Agoulmine, E. Lehtihet, “FOCALE – A Novel Autonomic Networking Architecture”, ITSSA Journal, Vol. 3, No. 1, May 2007, pages 64-79, ISSN 1751-1461

[11] J. Strassner, M. Ó Foghlú, W. Donnelly, N. Agoulmine, “Beyond the Knowledge Plane: An Inference Plane to Support the Next Generation Internet”, IEEE Global Information Infrastructure Symposium (GIIS 2007), 2-6 July, 2007, pages 112-119.

[12] J. Strassner, “DEN-ng Model Overview”, Joint ACF, EMANICS, and AutoI Workshop on Autonomic Management in the Future Internet, May 14, 2008

[13] M. Serrano, J. Serrat, J. Strassner, M. Ó Foghlú, “Management and Context Integration Based on Ontologies, Behind the Interoperability in Autonomic Communications”, extended journal publication of the SIWN International Conference on Complex Open Distributed Systems, Vol 1, No. 4, July 2007

[14] J. Schonwalder, A. Pras, and J.-P. Martin-Flatin, “On the future of Internet management technologies”, IEEE Communications Magazine, vol.41, no.10, pp. 90-97, October 2003

[15] J. Strassner, “Autonomic Networking – Theory and Practice”, 20th Network Operations and Management Symposium (NOMS) 2008 Tutorial, Salvador Bahia, Brazil, 2008

[16] http://www.omg.org/mda

[17] J. Strassner, J. de Souza, S. van der Meer, S. Davy, K. Barrett, D. Raymer, S. Samudrala, “The Design of a New Policy Model to Support Ontology-Driven Reasoning for Autonomic Networking”, Journal of Network and Systems Management, Volume 17, Issue 1, March 2009

[18] J. Strassner, “Policy Based Network Management”, Morgan Kaufman, ISBN 1-55860-859-1

[19] A. Budanitsky, “Lexical semantic relatedness and its application in natural language processing” Technical Report CSRG390, University of Toronto, 1999

[20] http://wordnet.princeton.edu/